

PDF hosted at the Radboud Repository of the Radboud University Nijmegen

The following full text is an author's version which may differ from the publisher's version.

For additional information about this publication click this link.

<http://hdl.handle.net/2066/74830>

Please be advised that this information was generated on 2017-12-06 and may be subject to change.

TAAL OF TEKEN?

stijlen van mens-computer interactie.

Peter Desain

NICI, Universiteit Nijmegen
Postbus 9104
6500 HE Nijmegen

[Desain, P. (1994). Taal of teken, stijlen van mens-computer interactie [Language or sign, styles of man-computer interaction]. In A. A. J. Mannaerts, P. J. G. Keuss, & G. Ten Hoopen (eds.), Mens-Computerinteractie. 109-129. Lisse: Swets & Zeitlinger.]

De beginnende gebruikers van een nieuw systeem of computerprogramma vormen vaak snel een oordeel over de kwaliteit van de interactie met de computer. Zij kunnen al op het eerste gezicht een groot aantal ongemakken en onhandigheden opsommen. Of, bij een goede gebruikersinteractie, kunnen ze het gemak en de beheersbaarheid aangeven. Als het nu zo makkelijk is om systemen te beoordelen op hun gebruiksgemak, hoe komt het dan dat programmeurs en ontwerpers vaak systemen afleveren met een slechte interactiecomponent?

Laten we eens kijken naar de gebruikers die door veelvuldig gebruik experts zijn geworden. Als we hen vragen naar de kwaliteit van de interactie, blijken een aantal ongemakken totaal vergeten. Allerlei vreemde eigenschappen van het systeem zullen zelfs verdedigd worden. De gebruiker heeft zich blijkbaar aangepast aan de machine, de interactie is ingeslepen ('overlearned') en reflex-matig geworden. Zo zijn ontwerpers en programmeurs ook experts in hun eigen systeem en zien zij de onhandigheden niet meer. Observatie van beginnende gebruikers is dus een belangrijke methode om inzicht te krijgen in de interactie tussen mensen en computers. Daarmee kunnen we moeilijkheden en inconsistenties op het spoor komen en hypothesen vormen over het gemak van de interactie.

Vragen we echter aan de nieuwe, zich ergerende gebruikers hoe een en ander beter geconstrueerd zou kunnen worden, dan zullen zij een aantal ad hoc aanwijzingen geven voor veranderingen die vaak het systeem alleen maar ondoorzichtiger en inconsistenter maken. Blijkbaar is een algeheel overzicht van het systeem een voorwaarde voor het maken van een goed ontwerp van de gebruikersinteractie. Omdat systemen meestal bij stukjes en beetjes worden ontworpen en geprogrammeerd, is het niet vreemd dat het programmeurs en ontwerpers niet vaak lukt om een consistente gebruikers-interactiecomponent af te leveren.

In veel systemen zijn ad hoc toevoegingen verstoppt onder een positieve annotatie als "extra mogelijkheden" of "handige opties". Programma's voor de Macintosh bijvoorbeeld groeperen hun acties onder menukopjes aan de bovenrand van het scherm. Ad hoc toevoegingen worden vaak geplaatst in een menu met het aantrekkelijke kopje 'goodies' of 'extra's'. Dit heeft een positieve bijklank maar maakt het systeem moeilijker in het gebruik (welk menu bevat nu een bepaalde actie?).

Stel dat uit observatie en onderzoek van gebruikers en bestudering van het ontwerp een aantal alternatieven voor de interactie voort komen. We kunnen dan door middel van experimenten evalueren welke oplossing voor een interactieprobleem de beste is (het snelste in het gebruik, het gemakkelijkst te onthouden, het snelst te leren) voor een

bepaalde groep gebruikers met een bepaalde taak. Experimentele psychologie is goed uitgerust met methodes om achteraf dergelijke alternatieven te evalueren. Maar de zoekruimte van alle mogelijke oplossingen voor een ontwerp is vaak zeer groot: men kan niet alle mogelijkheden uitproberen en evalueren.

Dus naast observaties voor de hypothesevorming en experimenten voor de evaluatie daarvan is theorievorming nodig over de computergebruiker. Op theoretische gronden zijn dan voorspellingen te doen over de consequenties van ontwerpbeslissingen. Zoals aangegeven in het artikel van Ponsioen in deze bundel kunnen de ontwerpcriteria uit de verschillende theorieën nog wel met elkaar botsen, een moeilijk probleem voor de ontwerpers, maar het gebruik van die criteria is toch te verkiezen boven willekeurige keuzes met een evaluatie achteraf.

Theorieën over het laagste (bijna fysische) niveau van de interactie zoals bijvoorbeeld over de helderheid en kleur van het beeldscherm en de plaatsing van toetsen op een toetsenbord, zijn voorhanden in de ergonomie. Psychologische functies op hoger niveau zijn pas later een onderwerp van ergonomische studie geworden (Norman, 1986). Het veld van de psychologie dat zich met dergelijke theorievorming bezighoudt heet *cognitieve ergonomie*.

Theorievorming levert modellen van de gebruiker en van de interactie tussen de gebruiker en de computer. Hoewel het onderzoeksveld jong is, zijn er reeds een aantal van deze modellen voorgesteld in de literatuur, elk met een eigen gezichtspunt. Een groep modellen neemt bijvoorbeeld de formele beschrijving van de interface als uitgangspunt. Exponenten van deze benadering zijn de Command Language van Moran (1981) met zijn Grammar en het Virtual Protocol Model van Nielsen (1986). Een andere groep stelt de cognitieve processen bij de gebruiker centraal. Tot deze groep behoort Rasmussen met het Framework for Cognitive Task Analysis (Rasmussen, 1986). Ook Card, Moran en Newell met hun GOMS-model (Goals, Operators, Methods and Selection rules) en Norman met zijn model voor het User Centered System Design (Norman, 1986). In dit artikel gaan we uit van een model van de gebruiker en zullen een (wat aangepast) model van Norman gebruiken.

De interactie

Norman geeft een beschrijving van twee verschillende benaderingen van mens-computer interactie:

"There are two major metaphors for the nature of human-computer interaction, a conversation metaphor and a model world metaphor. In a system built on the conversation metaphor, the interface is a language medium in which the user and system have a conversation about an assumed, but not explicitly represented world. In this case, the interface is an implied intermediary between the user and the world about which things are said. In a system built on the model world metaphor, the interface is itself a world where the user can act, and that changes state in response to the user actions. The world of interest is explicitly represented and there is no intermediary between user and world. Appropriate use of the model world metaphor can create the sensation in the user of acting upon the objects of the task domain themselves" (Norman, 1986).

Deze stijlen van interactie: conversatie en modelwereld zullen we in dit artikel verder uitwerken en vergelijken. De *conversationele stijl* werd o.a. gebruikt in de eerste tekstverwerkers voor computerconsoles met papieruitvoer. Een expliciete representatie van de tekst werd niet aan de gebruiker getoond, en het systeem reageerde op

commando's van de gebruiker door het afdrukken van een teken van acceptatie of een foutmelding. De meest geavanceerde vorm van de conversationele benadering is het systeem waarbij de dialoog in natuurlijke taal gevoerd kan worden.

De *modelwereld stijl* werd gebruikt in de tweede generatie tekstverwerkers voor beeldschermen. Een expliciete representatie van de tekst werd aan de gebruiker op het scherm getoond, en verandering van toestand trad op doordat de gebruiker toetsen en knoppen indrukte en daarmee acties initieerde waarvan het resultaat meteen zichtbaar was. In sommige modelwereld interfaces worden de objecten grafisch voorgesteld als iconen (pictogrammen). De gebruiker kan met een aanwijsapparaat (muis of lichtpen) deze grafische objecten manipuleren (verplaatsen, vervormen, etc). Deze acties worden dan onmiddellijk afgebeeld op corresponderende acties in het domein (weggooien, vergroten). Deze methode heet *directe manipulatie* (Shneiderman, 1982). Sinds de introductie van de Apple Macintosh, waarin veel ideeën overgenomen zijn van de Xerox Star (Smith, 1982), is directe manipulatie een populaire methode geworden voor de interactie met de gebruiker, ook op kleine computers. De tekstverwerkers voor deze machines gebruiken directe manipulatie voor een deel van hun functies. Omdat bij het gebruik van een directe manipulatie interface de gebruikers kunnen gaan denken dat zij de objecten uit het domein zelf (regels, woorden, letters) manipuleren en niet een grafische afbeelding daarvan, noemen we deze interfaces doorzichtig of transparant.. Het gebruik van deze methode zal dan de aandacht (die men beter bij het domein zelf kan houden) weinig belasten. *Transparantie* is een belangrijk kenmerk en draagt veel bij tot het succes van directe manipulatie.

De directe manipulatie stijl is een geweldige stap vooruit gebleken in het denken over de mogelijkheden van mens-computer interactie. De kracht van deze methode is echter nog niet volledig benut, en bij de ontwikkeling daarvan zullen ook de beperkingen blijken. Sterke en zwakke kanten moeten worden onderzocht om aan te kunnen geven wanneer het gebruik van deze interfacemethode wel en niet zinvol is: directe manipulatie is niet de oplossing voor alle communicatie tussen mens en computer. Voor deze evaluatie moeten we een idee hebben wat er in de gebruiker omgaat als deze een bepaalde stijl van interactie gebruikt.

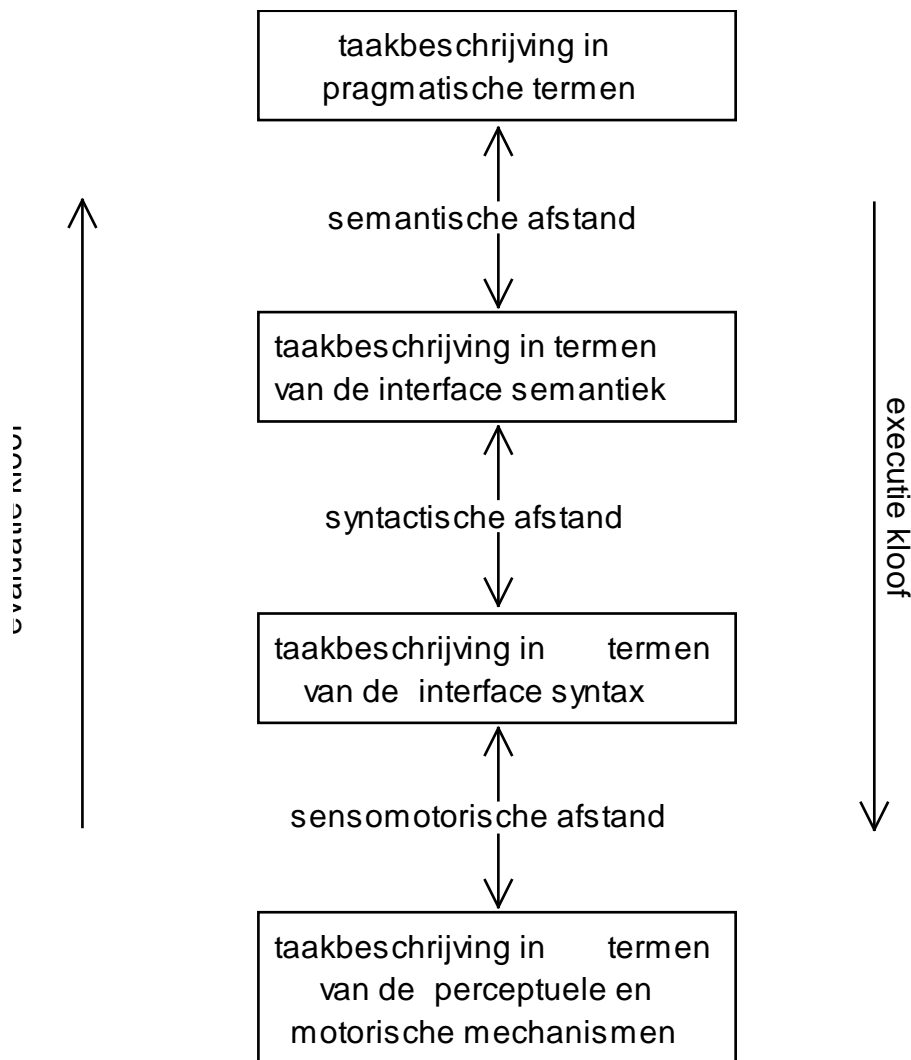
De gebruiker

Voor de gebruiker is de computer een middel om een bepaalde taak uit te voeren. De computer zal deze taak alleen op de juiste wijze uitvoeren als het correct bestuurd wordt met de juiste toetsaanslagen, bewegingen van de muis etc. De gebruiker zal echter de taak in een veel abstractere vorm, in termen van intenties en doelen geconceptualiseerd hebben. Er ligt een kloof tussen de intentie van de gebruiker en de uitvoering van de juiste handelingen die moeten leiden tot het bereiken van dat doel. De gebruiker zal dus een abstracte taakomschrijving moeten vertalen in concrete acties zoals toetsaanslagen. De afstand die overbrugd moet worden door deze vertaling noemt Norman (1986) de *executiekloof*. Andersom zal ook de uitvoer van de computer vertaald moeten worden van de gepresenteerde textuele of grafische vorm in het abstractere formaat om te begrijpen in hoeverre een actie succes heeft gehad. Deze vertaling overbrugt de zogenaamde *evaluatiekloof*. Norman postuleert een aantal niveaus van taakrepresentatie die d.m.v. vertalingen tussen representaties stapsgewijs de kloof overbruggen. Hieronder wordt een wat aangepaste versie van dit model beschreven (figuur 1).

De eerste taakrepresentatie geschiedt op een *pragmatisch niveau*.. De gebruiker heeft een intentie geformuleerd als een doel in termen van het taakdomein zelf (bijvoorbeeld ' ik wil een document weggooien'). Dit doel wordt vertaald in termen die betekenisvol zijn voor het formalisme van de interactiecomponent van het systeem (' ik wil het

bestand icon naar de prullenmand icon verplaatsen'). Dit noemen we het *semantisch niveau* van de taakrepresentatie. Deze representatie moet dan weer vertaald worden in termen die grammaticaal gezien een correcte invoer voor het interface zijn (' eerst de muis naar het bestands-icon verplaatsen, dan de muisknop indrukken, dan de muis naar het prullemand-icon verplaatsen, dan de muisknop loslaten'). Dit is het *syntactische niveau* van beschrijving. Als laatste moet deze representatie vertaald worden in de juiste coördinatiemechanismen van zintuigen en spieren. Dit laagste niveau noemen we het *sensomotorisch niveau*.

Natuurlijk vindt er ook een informatiestroom in de andere richting plaats. Van een sensomotorische ('een nieuw beeld op het netvlies') via een syntactische beschrijving ('er verdwijnt een bestands icon') en een semantische beschrijving ('er is een bestand weggegooid') naar een pragmatische ('ik ben mijn document nu kwijt'). Al deze omzettingen zullen bij een ervaren gebruiker bijna automatisch geschieden. En hoewel de eigenlijke interactie met de computer op het laagste niveau plaatsvindt, zal de ervaren gebruiker zich steeds minder bewust zijn van de informatieverwerkingsprocessen op de lagere niveaus (Nielsen, 1986).



Figuur 1. Representatie van taken op diverse niveaus

Aan de hand van de drie vertaaltappen van taakbeschrijvingen kan men drie maten geven voor de complexiteit van een bepaalde interactie. De maat die we hier 'afstand' zullen noemen, geeft de moeilijkheidsgraad weer van het vertalen van de ene in de andere representatie. De *semantische afstand* geeft aan hoe moeilijk het is om met het systeem te doen wat je zou willen en hoe moeilijk het is om te beoordelen of het systeem inderdaad heeft gedaan wat je wilt. Is de afstand groot dan is het zelfs voor experts niet mogelijk om het systeem voor een bepaald doel te gebruiken.

In een experiment met de IBM text editor XEDIT bleek dat de meeste gebruikers slechts een zeer klein deel van de functies van dit programma daadwerkelijk gebruikten (Rosson, 1983). Deze overbodige functionaliteit vermijden is een van de doelstellingen van een goed ontwerp van de semantiek van een interface. Hierbij moet wel opgemerkt worden dat overbodige functionaliteit de semantische afstand niet hoeft te vergroten als de vaak gebruikte functies op zich een consistent geheel vormen.

De *syntactische afstand* geeft aan hoe ingewikkeld het is om een gewenste en mogelijke actie met de interface uit te voeren en om de uitvoer van het systeem te herkennen en te interpreteren in termen van de betekenis daarvan. Als de syntactische afstand groot is dan zullen gebruikers een interface onhandig en lastig vinden.

In een experiment van Ledgard (1980) werd de commando syntax van twee tekst editors vergeleken. Hoewel beide editors commando's met dezelfde semantiek bevatten, werd in de syntax van de ene editor veel gebruik gemaakt van speciale symbolen en in de syntax van de andere alleen van sleutelwoorden (bijvoorbeeld RS:/KO/,/OK/;* versus CHANGE ALL 'KO' TO 'OK'). Het bleek dat de laatste editor bij het meten van de gemaakte fouten en de snelheid van werken een duidelijk voordeel had. Maar ook hier bleek het verschil kleiner te worden naarmate de gebruikers meer ervaring met de editor hadden.

De *sensomotorische afstand* geeft aan hoe complex het is om gegeven de juiste invoer voor de computer deze ook daadwerkelijk te effectueren in de juiste motorische handelingen. Ook de complexiteit van de perceptieprocessen die zorgdragen voor de herkenning van de uitvoer van de computer valt onder deze term. De sensomotorische afstanden worden reeds langer bestudeerd in de ergonomie. Toch worden in de huidige technologie lang niet alle resultaten hiervan benut om het de gebruiker makkelijker te maken.

Zo is de tastzin in het alledaagse leven een betrouwbare bron van informatie die we gebruiken bij het herkennen, vastpakken en manipuleren van objecten. Dit zintuig blijft echter nog ongebruikt bij de huidige directe manipulatie systemen met een muis of lichtpen. Toch zou het handig zijn om bv. door middel van tactiele feedback op de vinger of hand te kunnen voelen dat een icon aangewezen wordt.

Het is moeilijk om het jonge en heterogene onderzoeksterrein van de cognitieve ergonomie overzichtelijk in kaart te brengen. De bruikbaarheid van een model zoals boven geschetst blijkt uit het gemak waarmee allerlei toegepast onderzoek in te delen is naar de afstand die ze poogt te verkleinen. In figuur 2 is dit als voorbeeld gedaan voor een aantal bekende onderzoeksdomeinen. We zullen deze hieronder kort omschrijven.

	executie	evaluatie	beiden
semantische afstand	Do What I Mean	What You See Is What You Get	metaforen
syntactische afstand	afkortingen	schermopmaak	menustructuren
sensomotorische afstand	invoerapparaten	schermcontrast	oog-hand coördinatie

Figuur 2. Indeling van onderzoeksvelden aan de hand van Norman's schema.

Do What I Mean (DWIM) faciliteiten werden voor het eerst aangeboden in een LISP programmeeromgeving (Teitelman and Masinter, 1981). Als er een fout optrad in het programma bijvoorbeeld door een typefout van de programmeur dan kon het systeem een aantal redelijke alternatieven voorstellen. Er is vrij veel domeinkennis (in dit geval kennis van programmeren in LISP) noodzakelijk om zinnige correcties voor te stellen.

Afkortingen van commando's blijken vooral voor experts voordeel op te leveren. Er is onderzoek gedaan (Ehrenreich en Porcu, 1982; Benbasat en Wand, 1984; Schneider, 1984) dat uitwijst dat het gebruik van een consistente strategie voordelen heeft. Strategieën die gebruikt kunnen worden bij de constructie van afkortingen zijn ondermeer het afbreken na de eerste paar letters, het weglaten van klinkers en het gebruiken van alleen de eerste en laatste letters.

Invoerapparaten bestaan er in grote variëteit. Naast de bekende muis zijn er voor het aanwijzen onder meer lichtpennen, trackballs, joysticks, aanraakschermen, oogbewegingsvolgers en zelfs een handschoenopnemer die de stand van hand en vingers volledig aan de computer doorgeeft (Foley, 1987). Hoewel het soms verleidelijk is om onderzoeksresultaten te interpreteren in termen van de kwaliteit van het invoerapparaat, blijkt elk apparaat geschikt te zijn voor bepaalde taken en minder voor andere, zodat een zorgvuldige afweging steeds nodig blijft (Buxton, 1986).

What You See Is What You Get (WYSIWYG). Toen grafische beeldschermen met een hogere resolutie op de markt kwamen, werden de teksteditors voor deze systemen aangepast om de afbeelding van een tekst op het scherm precies zo weer te geven als de afgedrukte versie er uit zou zien. Dit verminderde de complexiteit van het bedienen van deze tekstverwerkers omdat de layout onafhankelijk werd van het uitvoermedium. Later kreeg de term WYSIWYG een bredere betekenis die in het algemeen aangeeft dat er in een directe manipulatie omgeving geen verborgen (niet grafisch gepresenteerde) domeinconcepten mogen zijn.

Schermopmaak kan, naast kennis over het specifieke karakter van beeldschermen als medium voor grafiek, gebruik maken van algemene principes van het grafisch ontwerpen. Bij het ontwerp van de Star interface moesten er iconen bedacht worden voor objecten als files en printers (Verplank, 1987). Deze objecten moesten geselecteerd getoond kunnen worden door het icon te inverteren (wit wordt zwart en vice-versa), waarbij een geselecteerd icon een duidelijke attentiewaarde heeft. Er werden groepen van iconen getest waarbij duidelijk bleek dat de iconen ieder overwegend wit moesten zijn om de

aandacht op het (nu overwegend zwarte) geselecteerde icon te richten. Verder bleek het noodzakelijk om de kopregels van vensters die overwegend zwart waren te veranderen in een lichter grijspatroon om goed onderscheidbaar te zijn van het geselecteerde icon.

Schermincontrast is een van de factoren die genoemd worden bij onderzoeken naar visuele vermoeidheid bij beeldschermwerkers (Padmos, 1988). Andere factoren zijn de verlichting en spiegelingen, de symboolgrootte en het gebruik van kleuren (De Weert, 1988). Hoewel er veel aanbevelingen en normen op dit terrein zijn die het bestaan van objectieve criteria doen veronderstellen, blijkt vaak dat de gemelde vermoeidheid niet los te zien is van een eenzijdige en inspannende taakstructuur.

Metaforen geven de gebruiker een concrete analogie van het systeem in een ander domein. Dit kan het leren gebruiken van een systeem soms versimpelen. Er is onderzoek gedaan naar de effectiviteit van zowel verbale als visuele metaforen (Van der Veer en Felt, 1988) waaruit bleek dat deze afhingen van de voorkeur voor verbale of picturale informatieverwerking van de individuele gebruiker. In de bijdrage van van de Veer aan deze bundel wordt ingegaan op de diverse aspecten van het gebruik van metaforen in mens-computer interactie.

Menu's worden in verschillende opeenvolgingen gebruikt. De meest gebruikte structuren zijn lineair (een aantal menus wordt in vaste volgorde doorgewerkt) en hiërarchisch (afhankelijk van de keuze van een menu verschijnt een ander menu). Als er nu uit een groot aantal mogelijke opties gekozen kan worden, kunnen die in een diepe hiërarchie (veel menu's met ieder weinig opties) of in een brede hiërarchie (weinig menus met ieder veel opties) gerangschikt worden. Vaak worden getallen rond 7 genoemd als optimum voor het aantal opties per menu (Miller, 1981).

Oog-hand-coördinatie is simpeler en sneller bij het gebruik van directe aanwijsapparaten zoals aanraakscherm en lichtpen. Deze zijn echter minder accuraat, geven aanleiding tot vermoeidheid van arm en schouder en kunnen een deel van het scherm onzichtbaar maken (Stammers en Bird, 1980). Indirecte apparaten zoals de muis hebben deze nadelen niet, maar geven een grotere mentale belasting.

De verschillende onderzoeksvelden en hun indeling in het schema (Figuur 2) illustreren het volgende: Het doel van een goed ontwerp van de mens-computer interactie is het verkleinen van de verschillende afstanden binnen de noodzakelijke functionaliteit die het programma moet aanbieden. Als we spreken over *directe interfaces* bedoelen we interfaces waarbij deze afstanden inderdaad klein zijn. Dit is in principe op twee manieren te bereiken. Ten eerste door gebruik te maken van bij de gebruiker reeds aanwezige capaciteiten voor vertaling tussen de verschillende taak-niveaus. Dit noemden we transparantie. De tweede methode maakt gebruik van het feit dat de computer de gebruikers tegemoet kan komen door het interpreteren van hun acties, infereren van hun doelen en de interactie daaraan aan te passen. Kempen (mondelinge mededeling) stelt hiervoor de term *perceptiviteit* voor. Beide methoden resulteren in een grotere directheid van de interface. Deze directheid kan alleen geëvalueerd worden voor een bepaalde groep gebruikers in een bepaald stadium van ervaring met het systeem en met een bepaalde taak. Zeer omslachtige en indirecte interactie kan toch direct worden als de gebruikers voor lange tijd gedwongen zijn deze te gebruiken, zich aanpassen aan de computer en zich de complexe vertalingen van taakniveaus aanleren.

Misschien het beste voorbeeld van dit fenomeen is het gebruik van de standaard QWERTY toetsenbord-layout. Deze is ontworpen om de letterhamertjes van oude mechanische schrijfmachines bij snel typen niet tegen elkaar te laten slaan. Toch zal een nieuwe layout gebaseerd op de eigenschappen van de menselijke gebruiker niet populair kunnen worden omdat de oude layout zo direct is geworden door het frequente gebruik.

In de volgende paragrafen zullen we de verschillende geïntroduceerde begrippen gebruiken om de sterke en zwakke kanten van directe manipulatie te beschrijven. We zullen daarbij alleen op de semantische en syntactische afstand letten. Voor vragen over de sensomotorische afstand verwijzen we verder naar Buxton (1986).

DE EVALUATIEKLOOF

Een mentaal model van het systeem

In een conversationele situatie moeten de gebruikers een model van het systeem construeren en bijstellen, anders kunnen ze het systeem niet goed gebruiken. Eerst, tijdens het leren omgaan met het systeem, moeten de regels ontdekt worden die aan het systeem ten grondslag liggen. Later, in het gebruik, zal steeds de momentane interne toestand van het systeem afgeleid moeten worden.

Door de gebruikers een metafoor aan te bieden kunnen objecten, mogelijke acties en consequenties daarvan al voorzien worden door gebruiker. De bekende Desktop metafoor geeft de gebruikers bijvoorbeeld de beschikking over objecten die zich min of meer gedragen als bekende zaken zoals hangmappen, prullemanden, wekkers en prikborden. Hoewel het systeem vaak sneller is te leren als het een metafoor gebruikt, is het dikwijls moeilijk om de analogie consistent vol te houden. Dit te meer daar een programma vaak extra functionaliteit aan kan bieden die niet meer in de analogie past. Een 'magische' toevoeging aan de metafoor is dan onontkoombaar (Smith, 1987).

Zelfs bij gebruik van een metafoor zal het mentale model dat de gebruikers van het systeem hebben niet compleet zijn en fouten bevatten. Om toch een zinnige conversatie aan te gaan zal de computer iets over deze misconcepties moeten weten. Deze kennis is niet expliciet voorhanden, maar zal afgeleid moeten worden uit de reacties van de gebruiker. De situatie is symmetrisch: ook de computer zal een intern model van zijn gesprekspartner moeten opbouwen. Of beter nog: een model van het model dat de gebruiker van de computer heeft. Dit is een van de meest moeilijke taken bij het ontwerpen van een dialoogsysteem.

In een directe manipulatie interface is een model van het systeem in de vorm van een plaatje op het beeldscherm altijd voorhanden voor de gebruiker. Van deze representatie kan de gebruiker veel overnemen voor zijn of haar mentale model. En omdat het systeem nu ook simpele hypothesen kan maken over wat de gebruiker weet (nl. wat op het scherm zichtbaar is), is het expliciet opbouwen van een model van de gebruiker minder noodzakelijk. Natuurlijk moet de grafische representatie van het domein dan wel goed genoeg zijn. De eisen die er aan een grafische representatie gesteld kunnen worden zodat die een goede basis vormt voor het mentale model, beschrijven we in de volgende paragrafen.

Representatie van objecten en toestanden

In een interfaceontwerp zullen goede keuzen gemaakt moeten maken voor representatie van individuele objecten door iconen of woorden.

Het blijkt dat plaatjes vaak beter onthouden worden dan woorden in veel verschillende taken (Paivo, 1976). Dus voor taken waarbij een groot aantal objecten onderscheidbaar gepresenteerd moet worden lijken iconen het geheugen minder te belasten.

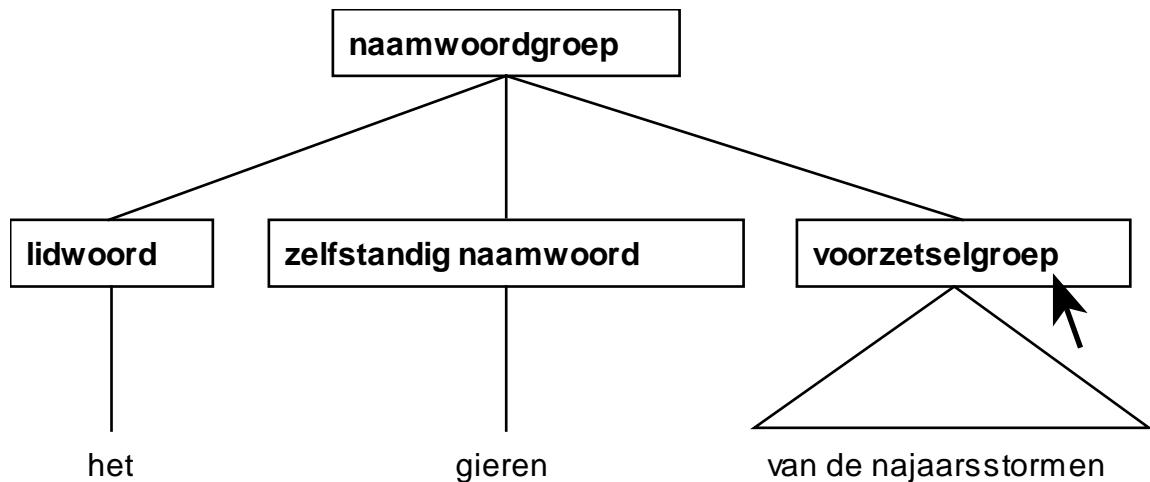
Deze verzamelingen van iconen zullen als consistent geheel ontworpen moeten worden, om herkenning te vergemakkelijken.

In een bepaald tekenprogramma voor de Macintosh wordt een pen en papier metafoor gebruikt. Alle acties en objecten zijn gerepresenteerd als bekende voorwerpen zoals pennen, gummetjes, inktpotten etc. en gedragen zich ook min of meer als hun equivalenten uit de realiteit. Er wordt echter ook gebruik gemaakt van een 'lasso' icon dat buiten deze metafoor valt, en dus op het eerste gezicht onbruikbaar is. Een overzicht van de moeilijkheden bij het ontwerp van een goede set pictogrammen voor mens-computer interactie is te vinden in Gittins (1986).

Naast individuele iconen en woorden drukken constellaties van deze objecten hun onderlinge relaties uit. Grafische en talige representaties verschillen in het gemak waarmee ze in zulke constellaties bepaalde soorten informatie kunnen weergeven. Grafische representaties zijn beter voor het weergeven van *structurele informatie* zoals schema's, bouwtekeningen en organisatiestructuren. *Functionele informatie* zoals de rol van een onderdeel in een electronicschema of het doel van een afdeling in een organisatie laat zich beter uitdrukken in taal. Om de semantische afstand niet te groot te laten worden moet er een modus gekozen worden die aansluit bij de soort informatie die overgedragen moet worden.

Vaak is het mogelijk om te kiezen of bepaalde informatie als geïsoleerd symbool (icon, woord) gepresenteerd wordt, of dat er meer uitleg, detail en interne structuur vermeld wordt. In interactie met een gebruiker zal het systeem dan ook altijd moeten kiezen in hoeverre deze details nodig zijn. In menselijke interactie bestaat de conventie dat alleen relevante informatie gegeven mag worden, dit wordt door Grice (1975) het Maxim of Quantity genoemd. Voor talige interactie bestaan daarvoor verschillende mechanismen: modificerende bijzinnen, al dan niet definite omschrijvingen en gebruik van parafrases. In grafische systemen is het moeilijk om alle relevante informatie uit het domein weer te geven en toch de gebruiker niet te belasten met een volgepakt scherm. Een oplossing is om informatie bij elkaar te nemen in afgesloten blokken, die dan door de gebruiker of het systeem te openen zijn als de informatie relevant wordt. Als het systeem goed ontworpen is, wordt hiermee ook *incrementeel leren* mogelijk.

In een directe manipulatie editor voor ontleedstructuren, die mede ontwikkeld werd voor een onderwijscontext (het leren ontleden van nederlandse zinnen op de middelbare scholen), worden de deelontledingen die teveel in detail zouden treden voor het huidige niveau van de leerling 'verstopt' in een afgesloten structuur (Desain, 1988). Dit symbool (een driehoek) is dan later weer te openen om de interne structuur te zien, zie figuur 3.



Figuur 3. Verbergen van details in een 'afgesloten' substructuur in een grafische editor voor ontleedbomen.

Als de grafische representatie noodgedwongen toch zo groot is dat alleen een deel op het scherm getoond kan worden, dan zijn er faciliteiten nodig om de gebruiker te helpen bij het navigeren door de structuur, zoals bladeren (browsing), verschuiven (scrolling) en schalen (zooming). Deze faciliteiten kunnen echter niet voorkomen dat er een grote geheugenbelasting optreedt als de gebruikers moeten onthouden waar bepaalde objecten zich op het virtuele scherm bevinden. Een overzicht in een apart venster, of minstens een signaal dat er zich aan bepaalde zijden buiten het scherm nog meer objecten bevinden is dan geboden. De mogelijkheid van een overzicht valt in de algemenere klasse van meervoudig perspectief (*multiple views*). Interfaces die deze methode ondersteunen geven de gebruikers de kans om op verschillende manieren (vaak ieder met een eigen venster) dezelfde informatie te bekijken. Deze gezichtspunten kunnen ook zeer abstract zijn en bijvoorbeeld slechts bepaalde relaties tussen objecten tonen. Als het mogelijk is om door middel van directe manipulatie de grafische representatie vanuit één perspectief te modificeren, dan is het natuurlijk noodzakelijk om alle andere representaties consistent te houden en waar nodig te veranderen.

Representatie van toestandsveranderingen

Waarheidsgetrouwheid is een noodzakelijke conventie voor menselijke communicatie. Dit wordt uitgedrukt in Grice's Maxim of Quality. Ook een grafische representatie van een domein in een directe manipulatie interface moet een waarheidsgetrouwe (gelijkvormige) afbeelding zijn van het domein zelf. We noemen zo'n gelijkvormige afbeelding een *isomorfisme*. Het voldoet namelijk aan de volgende eis. Als er in het domein een actie mogelijk is die een toestand *a* in een toestand *b* transformeert, dan moet in de interface een corresponderende actie bestaan die de representatie van toestand *a* omzet in de representatie van toestand *b*. Pas als een grafische representatie aan deze eis voldoet zal de transparantie nooit 'kapotgaan'. Een isomorfe afbeelding houdt ook in dat alle relevante concepten (die de uitkomst van een actie kunnen beïnvloeden) een grafische weergave hebben. Als er geen verborgen domeinconcepten bestaan zal dit het begrip van de gebruiker zeer vergroten. Dit is de bredere betekenis van de slogan 'What You See Is What You Get'. Hoewel deze regel eigenlijk voor de hand ligt wordt deze toch in een aantal interfaces geschonden.

Veel teksteditoren voor beeldschermen gebruiken een typemachine-metafoor. Hierbij dient de spatietoets alleen om de plek op het papier waar geschreven gaat worden - de cursor - een positie op te schuiven. Maar in deze editors wordt de spatie ook fysiek als ingetypt karakter opgeslagen in de interne tekstrepresentatie, zelfs als ze direct voor een einde-van-de-regel toets worden aangeslagen, waar ze onzichtbaar zijn voor de gebruiker. Nu zou dit geen probleem zijn als ze ook geen verdere invloed op het verloop van de acties zouden hebben. Maar vaak plaatst bijvoorbeeld het 'ga-naar-einde-van-vorige-regel' commando wel de cursor achter de serie onzichtbare spaties die aan het einde van de regel ingetypt waren. Dus het resultaat van dit commando wordt voor de gebruiker onvoorspelbaar. De domein-actie 'ga-naar-einde-van-vorige-regel' resulteert in een afbeelding die niet de verwachte representatie is van de interfaceactie toegepast op de interfacetoestand. Als we een typemachine-metafoor willen blijven gebruiken dan zou de oplossing kunnen zijn om alle acties blind te maken voor dit soort spaties. Zijn we bereid de metafoor te schenden dan is het zichtbaar maken van spaties een oplossing. Een zelfde probleem duikt overigens op bij teksteditoren die het einde van een document midden op het scherm kunnen zetten, maar geen onderscheid tonen met de situatie waarbij er een aantal lege regels de onderste helft van het scherm vullen. In het laatste geval is het mogelijk om met de 'ga-naar-volgende-regel' commando's de cursor te verplaatsen, in het eerste geval kan dat niet. Zorgvuldige aandacht voor dergelijke kleine details is bij het ontwerpen van gebruikers interfaces een absolute noodzaak.

Om in een directe-manipulatie interface steeds een waarheidsgetrouwe afbeelding van het domein aan de gebruiker te presenteren moet er zelfs tijdens toestandsveranderingen die geïnitieerd zijn door acties van de gebruiker of het systeem, een consistente grafische representatie onderhouden worden. Deze eis van *temporele continuïteit* betekent soms dat animatie noodzakelijk is. Dan hoeft de gebruiker zich niet te heroriënteren na een verandering ('waar was ik ook al weer').

Als in de Macintosh een venster gesloten wordt, wordt deze actie weergegeven door middel van de animatie van een zich verkleinende omkadering van het venster dat zich 'terugtrekt' in het icon dat het venster symboliseert.

De cognitieve kanten van deze problemen worden behandeld in Woods (1983) in zijn zogenaamde theorie van 'Visueel Momentum'. De eisen van temporele continuïteit zijn een belangrijke technische moeilijkheid bij de implementatie van directe manipulatie interfaces.

De executiekloof

Nadat we de grafische representatie van een domein in een directe manipulatie omgeving behandeld hebben, gaan we vervolgens beschrijven hoe de gebruiker daarmee handelingen kan uitvoeren.

Referentie

In veel conversationele interfaces is de taal waarin de gebruikers opdrachten geven anders dan de taal van de respons van de computer. Het is dan ook niet mogelijk om te refereren aan de oude uitvoer van het systeem om nieuwe invoer te construeren. Natuurlijk is dat wel zo bij het gebruik van natuurlijke taal. Daar kan de gebruiker

refereren aan de vorige uitingen van het systeem en van zichzelf. Maar het juist behandelen van zulke anaforische referenties is een groot probleem.

'Gooi het nu maar weg' is een uiting die goed begrepen zal worden in een intermenselijke communicatie. Voor een systeem dat moet beslissen wat er met 'het' bedoeld wordt is het echter moeilijk om uit de mogelijke kandidaten uit de vorige uitingen de juiste te selecteren. In natuurlijk taalgebruik worden aspecten als recentheid, focus, nadruk etc. gebruikt om zulke verwijzingen op te lossen (Hirst, 1981).

Ook in commandotalen voor beheerssystemen (operating systems) worden soms incomplete referenties ondersteund zoals *defaults* en *wildcards*. Defaults zijn standaard gegevens die het systeem gebruikt als de gebruiker een onvolledige specificatie geeft. Wildcards zijn een soort 'jokers' waarmee de gebruiker aangeeft dat elk gegeven dat gevonden kan worden acceptabel is. Zelfs in deze relatief simpele context is het gebruik van deze mechanismen een onderschat probleem, dat zichtbaar wordt door een incorrecte of onverwachte respons van het systeem.

In directe manipulatie interfaces zijn grafische representaties gebruikt voor invoer en uitvoer. Draper (1986) karakteriseert deze zogenaamde inter-referentialiteit als een belangrijke verschuiving in ons denken over computerinvoer en -uitvoer. Als deze interfaces goed ontworpen zijn is het altijd mogelijk om te refereren aan een bepaald object door het aan te wijzen. Dus het oplossen van anaphora is minder problematisch. Wat betreft het syntactische niveau moeten alle objecten waaraan gerefereerd kan worden aanwijsbaar zijn. Dit geldt ook voor abstracte objecten. In een conversationeel systeem is de mogelijkheid van *abstractie* vaak beschikbaar. Zoals we over concrete objecten ('deze file') kunnen praten, kunnen we immers ook refereren aan onbepaalde objecten ('een file'), klassen van objecten ('tekstfiles'), gekwantificeerde objecten ('alle files, gemaakt sinds gisteren'), genegerde objecten ('niet die file') en zelfs imaginaire objecten ('de file die ik morgen zal maken', 'de file die uit dit programma had moeten komen'). In grafische interfaces is abstractie vaak niet gerepresenteerd. Het icon voor het onbepaalde concept 'een file' bestaat bijvoorbeeld niet. Dat dit krachtige mechanisme afwezig blijft is een serieus nadeel van grafische interfaces. Ook acties zijn vaak niet expliciet gerepresenteerd, zodat een referentie aan een vorige actie niet mogelijk is ('doe het nu nog een keer'). Voor al deze problemen lijkt een overgang naar de conversationele methode de enige mogelijkheid.

Modes

Referentie aan objecten maakt het mogelijk om aan te geven dat er op een object een bepaalde actie kan worden toegepast. De gebruiker geeft dit aan met een commando. Deze commando's moeten voldoen aan een voorgeschreven syntax. Zelfs in directe manipulatie interfaces, waar alle acties direct onder de muisknop lijken te zitten is er nog heel wat onderzoekwerk nodig om de muissyntax te minimaliseren. Elke voorgeschreven syntaxregel zal immers zogenaamde modes introduceren. In een *mode* is slechts bepaalde invoer geldig en er wordt van de gebruiker voor die mode specifieke handelingen geëist.

In een systeem waarbij acties uit een menu te kiezen zijn kan elke actie een bepaald extra gegeven, een argument, nodig hebben. Dit is bijvoorbeeld het object waarop de actie moet plaatsvinden, de plaats waarheen iets verschoven moet worden, of het woord waarnaar gezocht moet worden. Na het kiezen van de actie komt de gebruiker

dan vaak in een 'geef-een-argument-voor-deze-actie' modus met een ander gedrag van het interface. Er kan dan slechts aan bepaalde typen objecten gerefereerd worden. Dit is vooral hinderlijk als er geen teken van het binnengaan van deze mode gegeven wordt en de gebruikers maar moeten raden wat er van hen verwacht wordt. Sommige systemen doen het andersom. Daarin moeten de gebruikers eerst het argument aangeven en dan de actie die uitgevoerd moet worden. De vraag is welke interactie methode beter is. Een oplossing voor dit 'comando eerst versus argument eerst' debat wordt gegeven door Lieberman (1985). Hij stelt voor om de volgorde vrij te laten. Het wordt pas echt ingewikkeld als een actie meerdere argumenten vraagt (het verbinden van twee objecten), niet alle acties op alle objecten kunnen plaatsvinden of er default (standaard) argumenten zijn voor bepaalde acties (bijvoorbeeld het laatst gegeven argument). Een mogelijke oplossing voor deze problemen is het selecteerbaar maken van alle relevante objecten die als argument kunnen dienen (dus ook de concepten van hogere orde zoals het tweetal objecten dat verbonden kan worden). Door bij elke type van object een specifiek menu van acties aan te bieden, is het mogelijk om het kiezen van niet-toepasselijke acties te voorkomen.

Soms is het mogelijk om modes en hun beëindiging directer of 'natuurlijker' te maken.

In de zogenaamde 'pull down' menus van de Macintosh is selectie van een menu en de selectie van een item daaruit tot één muisactie samengetrokken. Na selectie van het menukopje drukt men op de muisknop, het menu verschijnt, men beweegt de muis naar het gewenste item en laat de muisknop los. Zo is het verlaten van een mode ('kies item uit menu') natuurlijk gemodelleerd als het stoppen van waar je bezig bent: de knop loslaten (Buxton, 1986).

Ook zijn er andere oplossingen voor het aangeven van acties met de muis. De muis kan meerdere knoppen hebben die elk een eigen actie triggeren. De iconen zelf kunnen voorzien zijn van subregionen voor bepaalde acties (knoppen en hendels). Muisbewegingen kunnen herkend worden en bepaalde acties triggeren (bewegen rond een icon betekent roteren, een uitveegbeweging betekent weggooien etc.). Deze interactiemethode zal bij de toekomstige pencomputers essentieel worden.

Accuratesse

In conversatie is het mogelijk om elk gewenst niveau van precisie te gebruiken (niet zo groot, groter dan dat, ongeveer 3 meter, 4.1 milimeter). Voor grafische interfaces moeten daartoe geëigende methoden ontworpen worden. Er zijn soms mogelijkheden voor quantisatie, zoals het gebruik van een rooster. Voor de indirect gekoppelde aanwijsapparaten (de muis, in tegenstelling tot het aanraakscherm) is het mogelijk om de gevoeligheid in te stellen. Samen met een goede terugkoppeling over het aangewezen punt, kan dit de accuratesse vergroten.

Minder voor de hand liggend is de noodzaak van verminderde accuratesse. In natuurlijke taal is 'ongeveer hier' goed uit te drukken. In directe manipulatie heeft men behoefte aan het herkennen van bijvoorbeeld handbewegingen die vaak de uiting 'ongeveer hier' vergezellen. Er zijn verschillende vormen van aanwijzen (deixis) die inderdaad voor mens-machine-interactie aangewend kunnen worden (Rheithinger, 1987). Pagina opmaak zou bijvoorbeeld aan de accuratesse van het systeem overgelaten kunnen worden binnen de grenzen van de aanwijsbeweging van de gebruiker.

Fouten

Elke soort interface moet fouten in de invoer van de gebruiker op een of andere manier kunnen opvangen. De fouten die de gebruiker op het pragmatische niveau maakt zijn door het systeem niet te detecteren. De gebruiker geeft immers betekenisvolle en syntactisch correcte opdrachten aan het systeem, die ze zal uitvoeren. De gebruiker zal zich dan geconfronteerd zien met een toestand van het systeem die eigenlijk niet gewenst was ('de weggooide file bleek toch nog nodig te zijn'). In dergelijke situaties is het handig als er een faciliteit bestaat om een actie te 'ontdoen' en een oude toestand van het systeem te herstellen. Natuurlijk is dit alleen mogelijk in domeinen waar de actie zelf ook omkeerbaar is (en bijvoorbeeld niet in het domein van de computer-ondersteunde oorlogsvoering).

In een conversationeel systeem worden alle acties uitgevoerd in een niet expliciet gerepresenteerde context. Een gedane actie ongedaan maken is dan ook moeilijk voor de gebruiker omdat de actie vaak afhankelijk is van de context.

In natuurlijke taal zijn er een aantal mechanismen voorhanden om correcties aan te geven (zoals 'Ik bedoel', 'Ehh', 'Sorry, ..'). Spraakinterfaces, zoals beschreven in deze bundel in het hoofdstuk van van Nes, zullen deze uitingen goed moeten interpreteren. En de grammaticale component van dergelijke interfaces zal moeten worden voorzien van mechanismen die gedeeltelijk opgebouwde foutieve representaties weer weghalen (v. Wijk en Kempen, 1985).

In directe manipulatie interfaces zijn acties vaak makkelijk omkeerbaar (het leggen van een verbinding, het weghalen ervan). Ook 'undo' faciliteiten zijn makkelijker te gebruiken door representatie van de context. Dit kan resulteren in minder angst bij de beginnende gebruiker om acties uit te proberen en maakt een *vrije exploratie* mogelijk.

Fouten die de gebruiker op semantisch niveau maakt zijn vaak alleen door het systeem te ontdekken tijdens het uitvoeren van de gevraagde taak. Ze worden ook wel *dynamische fouten* genoemd. Doordat ze tijdens de verwerking optreden kunnen ze het systeem in een inconsistente toestand achterlaten. Door een zorgvuldig ontwerp van de syntax zijn soms dynamische fouten te veranderen in statische. *Statische fouten* zijn vóór de verwerking door het systeem te detecteren, doordat een syntaxregel geschonden is. Voor de overblijvende dynamische fouten kan de gebruiker soms de kans gegeven worden de toestand te herstellen en daarna door te gaan of opnieuw te starten met het verwerken vanaf een vorige toestand. Soms moet de verwerking afgebroken worden. In menselijke conversatie is een wijziging van context of interruptie een algemeen voorkomend fenomeen (Brown, 1983). Een procedure voor het afhandelen van een dynamische fout die de dialooggang even verstoort past goed in de conversationele metafoor. In model-wereld-interfaces is het optreden van dynamische fouten eigenlijk niet toegestaan omdat ze het karakter van de metafoor schenden. Deze fouten kunnen immers op elk moment optreden. Ze worden niet door de gebruiker zelf geïnitieerd. De gebruiker verliest de controle. Een wisseling naar een conversationele stijl is soms de enige oplossing.

Fouten die de gebruiker op een syntactisch niveau maakt (statische fouten) zijn door het systeem te detecteren vóórdat ze uitgevoerd zijn door de vorm van de invoer te controleren. De uiting kan dan geweigerd worden met opgaaf van een reden. Het vervaardigen van grammaticaal correcte invoer kan voor de gebruiker een moeilijke taak zijn tenzij de syntaxis simpel is of de computer hulp geeft om fouten te voorkomen. In bepaalde conversationele interfaces wordt de gebruiker gedwongen tot syntactisch correcte invoer doordat de woorden gekozen moeten worden uit menus waarin alleen

correcte voortzettingen staan van de reeds gedeeltelijk gekozen zin. Zo kunnen syntactische fouten worden voorkomen. Toch geeft dit een onnatuurlijk gebruik van taal waarbij het nog maar de vraag is of de gebruiker tóch niet goed de toegestane syntax moet kennen om de acties te plannen. In directe manipulatie interfaces is het mogelijk om op een natuurlijke manier de syntactische fouten te voorkomen, door bepaalde muisacties onmogelijk te maken.

In een editor voor linguïstische bomen voor het onderwijs werd het op deze manier onmogelijk gemaakt om foute verbindingen te leggen tussen grammaticale constituenten en bijvoorbeeld een lidwoord als zinsdeel aan een zin te verbinden in plaats van aan een naamwoordgroep (Desain, 1986).

Voor alle domeinen waarin sprake is van regels van syntactische correctheid zijn soortgelijke oplossingen mogelijk. Zij vormen een groot voordeel van grafische interfaces.

Conclusie

Directe manipulatie heeft een aantal voordelen als methode voor mens-computer interactie zoals het makkelijk ondersteunen van omkeerbaarheid, het voorkomen van statische fouten en het simpele gebruik voor structurele informatie. Het heeft ook een aantal zwakke punten. Vooral bij het gebruik van abstractie, het ondersteunen van verschillende niveaus van detail, de opvang van dynamische fouten en het behandelen van functionele informatie lijkt de conversationele methode superieur.

Een combinatie van interactiestijlen in een zogenaamd *multi-modale interface* ligt voor de hand. De optimale manier van samengaan van verschillende stijlen is een veelbelovend nieuw onderzoeksterrein.

Nawoord

Na het schrijven van dit artikel in 1988 hebben de ontwikkelingen uiteraard niet stil gestaan. Directe manipulatie interfaces zijn gemeengoed geworden voor veel applicaties. Multi-modale interface stijlen beginnen vanuit de laboratoria hun weg naar de gebruiker te vinden. En systematisch cognitief-ergonomisch onderzoek is niet meer weg te denken bij het ontwerp van informatiesystemen. Hoewel het genoemde onderzoek en de referenties dus niet meer volledig up to date te noemen zijn, is het geschetste theoretische kader nog geheel geldig en kan dienen om de nieuwste technologische ontwikkelingen en recent onderzoek en in perspectief te plaatsen.

Referenties

Badre, A., B. Shneiderman (1982) Directions in Human Computer Interaction.. New York: Ablex Publishers, Norwood.

Benbasat, I., Y.Wand (1984) Command Abbreviation Behavior in Human-Computer Interaction. Comm. of the ACM 27,4,376-383

Brown G., G. Yule (1983) Discourse Analysis. Cambridge University Press. Cambridge.

- Buxton W. (1986) There's More to Interaction Than Meets the Eye: Some Issues in Manual Input. In Norman D., S. Draper (Eds.) User Centered System Design: New Perspectives on Human Computer Interaction. Erlbaum, London.
- Card, S.K., T.P. Moran, A. Newell (1983) The Psychology of Human-Computer Interaction. Hillsdale NJ: Erlbaum.
- Clark, A.A. (1986) A Three-Level Human-Computer Interface Model. Int. J. Man-Machine studies 24, 503-517.
- Clark H., D. Wilkes-Gibbs (1986) Referring as a Collaborative Process. Cognition, 22, 1-39.
- Grice, H.P. (1975) Logic and Conversation. In P.Cole and J.L. Moran (Eds.) Studies in Syntax. Vol 3: Speech Acts. New York: Academic Press.
- Desain, P. (1986) Tree Doctor: a Software Package for Graphical Manipulation and Animation of Tree Structures. In G.C. van der Veer, G. Mulder (Eds.). Human-Computer Interaction: Psychonomic Aspects. Berlin: Springer-Verlag.
- Desain P. (1986) Graphical Programming in Computer Music, a Proposal. in P. Berg (Ed.) Proceedings of the International Computer Music Conference. CMA San Francisco.
- Ehrenreich, S. L., T. Porcu (1982) Abbreviations for automated systems: Teaching operators and rules, In Badre, A. & Shneiderman, B. Directions in Human Computer Interaction.. New York: Ablex Publishers, Norwood.
- Gittins D. (1986) Icon-based Human Computer Interaction. Int. J. Man-Machine studies 24, 519-543.
- Gittins D., R.L. Winder, H.E. Bez (1984) An Icon-driven End-user Interface to UNIX. Int. J. Man-Machine studies 21, 451-461.
- Hayes (1986) Steps toward Integrating Natural Language and Graphical Interaction for Knowledge-based Systems. Proceedings ECAI 1986.
- DiSessa, A. (1986) A Principled Design for an Integrated Computational Environment. Human Computer Interaction. 1,1 1-49.
- DiSessa, A. (1986) Notes on the Future of Programming: Breaking the Utility Barrier. In Norman D., S. Draper (Eds.) User Centered System Design: New Perspectives on Human Computer Interaction. Erlbaum, London.
- Draper S. (1986) Display Managers as the Basis for User-Machine Communication. In D. Norman, S. Draper (Eds.) User Centered System Design: New Perspectives on Human Computer Interaction. Erlbaum, London.

- Foley, J.D., V.L. Wallace, P. Chan. (1984) The Human Factors of Computer Graphics Interaction Techniques. IEEE Computer Graphics and Applications. 4, 13-48.
- Foley, J.D. (1987) Interfaces for Advanced Computing. Scientific American, October 1987.
- Kobsa A., J. Allgayer, C. Redding, M. Reithinger, D. Schmauks, K. Harbusch, W. Wahlster (1986) Combining Deictic Gestures and Natural Language for Referent Identification. Report, KI-labor, Universität des Saarlandes.
- Ledgard, H., J.A. Whiteside, A. Singer and W. Seymore (1980) The Natural Language of Interacting Systems. Comm. of the ACM 23, 556-563
- Levien R. (1986) Visual Programming. Byte, 11,2 135-114
- Lieberman H. (1985) There's More to Menu Systems Than Meets the Screen. Siggraph 19,3 ACM
- Miller, D.P. (1981) The Depth/Breadth Tradeoff in Hierarchical Computer Menus. Proceedings of the Human Factors Society, 25th annual meeting, 296-300.
- Norman D., S. Draper (Eds.) (1986) User Centered System Design: New Perspectives on Human Computer Interaction. Erlbaum, London.
- Padmos, P. (1988) Visual Fatigue with Work on Visual Display Units: The Current State of Knowledge. In G.C. van der Veer, G. Mulder (Eds.). Human-Computer Interaction: Psychonomic Aspects. Berlin: Springer-Verlag.
- Paivo, A. (1976) Imagery in Recall and Recognition. in J.Brown (Ed.) Recall and Recognition. New York: John Wiley and Sons.
- Rasmussen J. (1986) Information Processing and Human-Machine Interaction. Elsevier, New York.
- Reithinger N. (1987) Generating Referring Expressions and Pointing Gestures. In Kempen G. (Ed.) Natural Language Generation. Kluwer. Dordrecht.
- Rosson, M.B. (1983) Patterns of Experience in Text Editing. Proceedings of the CHI Conference on Human Factors in Computing Systems. 17-19.
- Shneider, M.L. (1984) Ergonomic Considerations in the Design of Text Editors. In Y. Vassiliou (Ed.) Human Factors and Interactive Computer Systems. Ablex Publishers, Norwood, NJ.
- Shneiderman B. (1982) The Future of Interactive Systems and the Emergence of Direct Manipulation. Behaviour and Information Technology, 1, 237-256.
- Shneiderman B. (1987) Designing the User Interface: Strategies for Effective Human-Computer Interaction. Reading, MA: Addison-Wesley.

- Smith D.C., C. Irby, R. Kimball (1982) The Star User-Interface: an Overview. Proceedings of the 1982 National Computer Conference.
- Smith, R.B. (1987) Experiences with the Alternate Reality kit, an Example of the Tension between Literalism and Magic. Proceedings of the CHI+GI conference, Toronto.
- Stammers, D.C. Bird, J.M. (1980) Controller Evaluation of a Touch Input Air Traffic Data System: An Indelicate Experiment. Human Factors 22,5,581-589.
- Teitelman, W., M. Masinter (1981) The Interlisp Programming Environment. Computer, April 1981, 25-33.
- Thomassen, A.J.W.M., H.L. Teulings, L.R.B Schomaker (1986) Real-time Processing of Cursive Writing and Sketched Graphics. In G.C. van der Veer, G. Mulder (Eds.) Human-Computer Interaction: Psychonomic Aspects. Berlin: Springer-Verlag.
- Vassiliou, Y. (Ed.) (1984) Human Factors and Interactive Computer Systems. Ablex Publishers, Norwood, NJ.
- Veer, G.C. van der, M. Felt (1988) Development of Mental Models of an Office System: A Field Study on an Introductory Course. In G.C. van der Veer, G. Mulder (Eds.). Human-Computer Interaction: Psychonomic Aspects. Berlin: Springer-Verlag.
- Veer, G.C. van der, G. Mulder (Eds.) (1988). Human-Computer Interaction: Psychonomic Aspects. Berlin: Springer-Verlag.
- Verplank B. (1987) Designing Graphical User Interfaces. CHI+GI tutorial, Toronto.
- Weert Ch.M.M. de (1988) The use of Color in Visual Displays. In G.C. van der Veer, G. Mulder (Eds.). Human-Computer Interaction: Psychonomic Aspects. Berlin: Springer-Verlag.
- Woods D.D. (1984) Visual Momentum: a Concept to Improve the Cognitive coupling of Person and Computer. Int. J. of Man-Machine Studies 21, 229-244
- Wijk van C., G. Kempen (1985) A Dual System for Producing Self-Repairs in Spontaneous Speech: Evidence from Experimentally Elicited Corrections. Report, Dept. of Psychology, University of Nijmegen.

Woordenlijst

cognitieve ergonomie

De studie van de mentale processen van gebruikers bij hun interactie met machines.

conversatieve stijl

Een taalachtige interactiestijl met de interface als intermediair tussen gebruiker en domein.

- default
Een standaardgegeven dat gebruikt wordt bij het ontbreken van verdere specificatie van de gebruiker.
- directe interface
Een gebruikersinteractie met simpele vertalingen tussen de taakrepresentaties op verschillende niveaus.
- directe manipulatie
Een interactie in modelwereldstijl met grafische representaties van domeinobjecten waarop de acties van de gebruiker aangrijpen door middel van bv. een muis.
- Do-What-I-Mean
Een methode voor opvang van typefouten in een programmeeromgeving waarbij contextinformatie gebruikt wordt om correcties voor te stellen.
- dynamische fout
Een fout in de invoer van de gebruiker die slechts te ontdekken is tijdens het verwerken daarvan.
- evaluatiekloof
De afstand die door de gebruiker overbrugd moet worden bij de constructie van een mentale representatie van de toestand van het domein op basis van de toestand van de interface.
- executiekloof
De afstand die door de gebruiker overbrugd moet worden bij het omzetten van domeinintenties in concrete interfacehandelingen.
- incrementeel leren
Het stapsgewijs eigen maken van een systeem met de mogelijkheid om het reeds te gebruiken terwijl de gebruiker slechts bekend is met een deel van de functionaliteit.
- isomorfisme
Gelijkvormigheid van gedrag van objecten in een domein en het gedrag van de corresponderende interface objecten.
- metafoor
De afbeelding van, uit een andere context bekende, mechanismen op het gedrag en de representatie van een interface.
- mode
Een tijdelijke toestand van de interface met specifieke eisen aan de invoer van de gebruiker.
- modelwereld stijl
Een interactiestijl waarbij het domein expliciet is gerepresenteerd in de interface en de handelingen van de gebruiker daar rechtstreeks op aangrijpen.
- multi-modale interface
Een interface die meerdere in- en uitvoer media en interactiestijlen consistent combineert.
- multiple views
De mogelijkheid om dezelfde data tegelijkertijd op meerdere manieren aan de gebruiker te presenteren.
- perceptiviteit
De actieve aanpassing van de interface aan de gebruiker door ondermeer het afleiden van gebruikersintenties.
- pragmatisch niveau
Een taakrepresentatie in termen van domeinintenties van de gebruiker.

semantisch niveau

Een taakrepresentatie in termen van de betekenis van interface objecten.

semantische afstand

De complexiteit van vertalingen tussen het semantische en het pragmatische niveau van taakbeschrijving.

sensomotorisch niveau

Een taakrepresentatie in termen van de betekenis van interface objecten.

sensomotorische afstand

De complexiteit van vertalingen tussen het sensomotorische en het syntactische niveau van taakbeschrijving.

statische fout

Een grammaticale fout in de invoer van de gebruiker die te ontdekken is voordat het programma die verwerkt.

syntactische afstand

De complexiteit van vertalingen tussen het syntactische en het semantische niveau van taakbeschrijving.

syntactische niveau

Een taakrepresentatie in termen van de vorm en grammatica van interface objecten.

temporele continuïteit

De eis dat voor een goede interactie plotselinge grote veranderingen op het scherm vermeden worden.

transparantie

De eigenschap van interfaces die zo direct zijn in het gebruik dat de gebruiker zich niet bewust is van de verschillende vertaaltappen tussen taakrepresentaties.

vrije exploratie

De mogelijkheid voor de gebruiker om al proberend een systeem te leren gebruiken.

What-You-See-Is-What-You-Get

Gelijkvormigheid van de grafische representatie op een beeldscherm en in de gedrukte vorm . Meer algemeen: het visualiseren van alle relevante domein informatie.

wildcard

Een symbool in de invoer van de gebruiker dat aangeeft dat elke willekeurige specificatie op die plek volstaat.